REMARKS

The Examiner's Action mailed on May 2, 2006, has been received and its contents carefully considered.

In this Amendment, Applicants have editorially amended Claims 1 and 18. Claims 1 and 18 are the independent claims, and claims 1-33 remain pending in the application. For at least the following reasons, it is submitted that this application is in condition for allowance.

Claims 1-6, 8, 13, 17-23, 25 and 30 were rejected under 35 U.S.C. §102(b) as being anticipated by *Maa* (US 5,878,057). This rejection is respectfully traversed.

Claim 1 recites the steps of "preparing a generator matrix representing a Linear Feedback Shift Register (LFSR) corresponding to a form for linearly mapping an input vector to a remainder vector; arranging the message inputted in the form to the input vector; multiplying the generator matrix to the input vector derived from the message; and producing a CRC result", and claim 18 recites "means for arranging the message inputted in a form to an input vector; a generator matrix representing a Linear Feedback Shift Register (LFSR) corresponding to the form for linearly mapping the input vector to a remainder vector; and means for producing a CRC result".

Maa discloses a highly parallel cyclic redundancy code (CRC) generator and method thereof to generate partial CRCs in parallel to any degree, in which a microprocessor precomputes the remainders that are loaded into a lookup table

(10), a CRC register (22) having a p-bit portion (22P) and a k-bit portion (22K), a control circuit (24) to control the CRC register (22), an input XOR gate (26) to process a data message and the contents of the p-bit portion (22P) to generate a result for storage in the p-bit portion (22P), and the content of the p-bit portion (22P) is used to control the output gates (12) to determine which of the remainders from the lookup table (10) are input to the parallel XOR tree circuit (14), the output of the parallel XOR tree circuit (14) is loaded into the k-bit portion (22K), the output of the k-bit portion (22K) is the partial CRC, the contents of the CRC register (22) are shifted to the left and the process repeated until the data message has been processed, and the contents of the k-bit portion (22K) are output as the CRC for the data message (col. 1, line 51 to col. 2, line 5; col. 4, lines 1-18; FIG. 2).

The Office Action asserts that the data message is arranged in the register (22) according to the input XOR gate (26), and the input data is arranged in the p-bit portion (22P) and the data in the p-bit portion (22P) is passed to an XOR tree (14) with a corresponding reminder from the lookup table (10) to produce a partial CRC.

It is respectfully submitted that this is not correct. In *Maa*, the input data message is inputted to the input XOR gate (26) together with the contents of the p-bit portion (22P), the output of the input XOR gate (26) writes back into the p-bit portion (22P), and the contents of the p-bit portion (22P) determine which the

remainders from the lookup table (10) are input to the parallel XOR tree circuit (14) to produce a partial CRC.

Furthermore, *Maa* utilizes the XOR and the parallel XOR to generate the CRC rather than utilizing linear feedback shift registers (LFSR) to generate the CRC, that is, *Maa* does not teach arranging the message inputted in a form to an input vector and to utilize a generator matrix representing a LFSR corresponding to the form for linearly mapping the input vector to a remainder vector to produce the CRC. In *Maa*, an iterative XOR process produces the CRC rather than an iteration procedure between the remainder vector and the input vector produces the CRC.

Therefore, *Maa* fails to teach or suggest the features of claims 1, 18 and 17 of the present application.

Dependent claims 2 and 19 recite "wherein the LFSR is configured for the message to be shifted thereinto from a MSB side", and dependent claims 3 and 20 recite "wherein the LFSR is configured for the message to be shifted thereinto from a LSB side".

In *Maa*, the control circuit (24) controls the CRC register (22) and the p most significant bits of the CRC register (22) in turn to control the gates (12) (col. 4, lines 5-6 and col. 3, lines 56-58).

The Office Action asserts that the p most significant bit (MSB) controls the register, and it is respectfully submitted that this misunderstands the technological

content of *Maa*. Besides, *Maa* cannot configure the LFSR because there is no LFSR in *Maa*.

Therefore, *Maa* fails to teach or suggest the features of claims 2-3 and 19-20 of the present application.

Claims 4-6 and 21-23 add limitations that the form is a byte-wise, word-wise or doubleword-wise form.

Maa notes to perform 64-bit CRC generation by existing CRC algorithms that are cumbersome because the degree of the generator polynomial is only 16 or 32 (col. 1, lines 34-38). *Maa* generates partial CRCs in parallel to any degree by producing a different set of per-computed remainders corresponding to each generator polynomial (col. 4, lines 26-30). In other words, *Maa* does not teach the message inputted in different form.

Therefore, *Maa* fails to teach or suggest the features of claims 4-6 and 21-23 of the present application.

Claims 8, 13, 25 and 30 claim initiating the LFSR with a specific value.

In Maa, the CRC register (22) is initialized to all zeros and the CRC computation does not require the length of the message (col. 1, lines 52-55; col. 3, lines 35-38; and col. 4, lines 60-61). But this application determines the initial value of the LFSR according to the length type of the message. *Maa* and this application are different.

Therefore, *Maa* fails to teach or suggest the features of claims 8, 13, 25 and 30 of the present application.

Claims 9-10, 14-16, 26-28 and 31-30 were rejected under 35 U.S.C. §103(a) as being obvious over *Maa* in view of *Mőrsberger* (US 6,560,746 B1). This rejection is respectfully traversed.

Mőrsberger provides a method and a parallel CRC generation circuit that decrease the hardware amount necessary for the parallel CRC generation circuit and the processing time by canceling internal state transitions that occur twice for the same bit (col. 3, line 29 to col. 4, line 7).

The Office Action asserts that increasing speed in computing CRC is an advantage when selecting the length type, and it is respectfully submitted that this misunderstands the technological content of *Mőrsberger*. Additionally, *Mőrsberger* points out clearly "the parallel generation circuit generates a CRC code of length N" and "the length N of the desired CRC code and the type of the CRC code is selected by choosing a value for N and by defining which of the coefficients a_n, n=0..., N-1 are 0 or 1"(col. 4, lines 60-61 and col. 5, lines 16-21), namely the N is the length of the CRC code rather than the length of the message.

That is to say, *Mőrsberger* fails to disclose identifying a length type of the message and determining the initial value of the LFSR in accordance with the length type. Moreover, in *Maa*, the CRC computation does not require the length of the message (col. 4, lines 60-61).

Therefore, *Maa* and *Mőrsberger*, whether taken separately or in combination, fail to teach or suggest the features of claims 9-10, 14-16, 26-28, and 31-33 of the present application.

It is submitted that this application is in condition for allowance. Such action and the passing of this case to issue are requested.

Should the Examiner feel that a conference would help to expedite the prosecution of this application, the Examiner is hereby invited to contact the undersigned counsel to arrange for such an interview.

Should any fee be required, however, the Commissioner is hereby authorized to charge the fee to our Deposit Account No. 18-0002, and advise us accordingly.

Respectfully submitted,

August 2, 2006
Date

Alun L. Palmer – Reg. No. 47,838
RABIN & BERDO, PC – Cust. No. 23995
Facsimile: 202-408-0924
Telephone: 202-371-8976

ALP/klc